



## Using bio.tools to generate and annotate workbench tool descriptions

Hillion, Kenzo-Hugo; Kuzmin, Ivan; Khodak, Anton; Rasche, Eric; Crusoe, Michael; Peterson, Hedi; Ison, Jon; Ménager, Hervé

*Published in:*  
F1000Research

*Link to article, DOI:*  
[10.12688/f1000research.12974.1](https://doi.org/10.12688/f1000research.12974.1)

*Publication date:*  
2017

*Document Version*  
Publisher's PDF, also known as Version of record

[Link back to DTU Orbit](#)

*Citation (APA):*  
Hillion, K-H., Kuzmin, I., Khodak, A., Rasche, E., Crusoe, M., Peterson, H., Ison, J., & Ménager, H. (2017). Using bio.tools to generate and annotate workbench tool descriptions. *F1000Research*, 6, [2074].  
<https://doi.org/10.12688/f1000research.12974.1>

---

### General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.


- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.



## SOFTWARE TOOL ARTICLE

# Using bio.tools to generate and annotate workbench tool descriptions [version 1; referees: 4 approved]

Kenzo-Hugo Hillion<sup>1</sup>, Ivan Kuzmin<sup>2</sup>, Anton Khodak<sup>3</sup>, Eric Rasche<sup>4</sup>,  
Michael Crusoe <sup>5</sup>, Hedi Peterson<sup>2</sup>, Jon Ison<sup>6</sup>, Hervé Ménager<sup>1</sup>

<sup>1</sup>Bioinformatics and Biostatistics HUB, Centre de Bioinformatique, Biostatistique et Biologie Intégrative (C3BI, USR 3756 Institut Pasteur et CNRS), Paris, France

<sup>2</sup>Institute of Computer Science, University of Tartu, Tartu, Estonia

<sup>3</sup>Igor Sikorsky Kyiv Polytechnic Institute, National Technical University of Ukraine, Kyiv, Ukraine

<sup>4</sup>Lehrstuhl für Bioinformatik, Institut für Informatik, Albert-Ludwigs-Universität Freiburg, Freiburg, Germany

<sup>5</sup>Common Workflow Language Project, Vilnius, Lithuania

<sup>6</sup>DTU Bioinformatics, Technical University of Denmark, Copenhagen, Denmark

**v1** First published: 30 Nov 2017, 6(ELIXIR):2074 (doi: 10.12688/f1000research.12974.1)

Latest published: 30 Nov 2017, 6(ELIXIR):2074 (doi: 10.12688/f1000research.12974.1)

## Abstract

Workbench and workflow systems such as Galaxy, Taverna, Chipster, or Common Workflow Language (CWL)-based frameworks, facilitate the access to bioinformatics tools in a user-friendly, scalable and reproducible way. Still, the integration of tools in such environments remains a cumbersome, time consuming and error-prone process. A major consequence is the incomplete or outdated description of tools that are often missing important information, including parameters and metadata such as publication or links to documentation. ToolDog (Tool DescriptiOn Generator) facilitates the integration of tools - which have been registered in the ELIXIR tools registry (<https://bio.tools>) - into workbench environments by generating tool description templates. ToolDog includes two modules. The first module analyses the source code of the bioinformatics software with language-specific plugins, and generates a skeleton for a Galaxy XML or CWL tool description. The second module is dedicated to the enrichment of the generated tool description, using metadata provided by bio.tools. This last module can also be used on its own to complete or correct existing tool descriptions with missing metadata.







This article is included in the **ELIXIR** gateway.

## Open Peer Review

Referee Status:

	Invited Referees			
	1	2	3	4
<b>version 1</b>				
published 30 Nov 2017	report	report	report	report

- 1 **Michael L. Heuer** , National Marrow Donor Program (NMDP), USA  
University of California, USA
- 2 **Christopher J. Fields** , University of Illinois at Urbana-Champaign, USA
- 3 **Manuel Corpas** , Cambridge Precision Medicine, UK
- 4 **Brian O'Connor** , University of California, Santa Cruz, USA

## Discuss this article

Comments (0)

**Corresponding authors:** Kenzo-Hugo Hillion ([kenzo-hugo.hillion1@pasteur.fr](mailto:kenzo-hugo.hillion1@pasteur.fr)), Hervé Ménager ([herve.menager@pasteur.fr](mailto:herve.menager@pasteur.fr))

**Author roles:** Hillion KH: Software, Writing – Original Draft Preparation; Kuzmin I: Software, Writing – Review & Editing; Khodak A: Software, Writing – Review & Editing; Rasche E: Software, Writing – Review & Editing; Crusoe M: Methodology, Software, Writing – Review & Editing; Peterson H: Funding Acquisition, Writing – Review & Editing; Ison J: Conceptualization, Funding Acquisition, Writing – Review & Editing; Ménager H: Conceptualization, Funding Acquisition, Software, Supervision, Writing – Original Draft Preparation

**Competing interests:** No competing interests were disclosed.

**How to cite this article:** Hillion KH, Kuzmin I, Khodak A *et al.* **Using bio.tools to generate and annotate workbench tool descriptions [version 1; referees: 4 approved]** F1000Research 2017, 6(ELIXIR):2074 (doi: [10.12688/f1000research.12974.1](https://doi.org/10.12688/f1000research.12974.1))

**Copyright:** © 2017 Hillion KH *et al.* This is an open access article distributed under the terms of the [Creative Commons Attribution Licence](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

**Grant information:** ELIXIR-EXCELERATE is funded by the European Commission within the Research Infrastructures Programme of Horizon 2020 [676559].

**First published:** 30 Nov 2017, 6(ELIXIR):2074 (doi: [10.12688/f1000research.12974.1](https://doi.org/10.12688/f1000research.12974.1))

## Introduction

Over the last few years, bioinformatics has played a major role in the field of biology, raising the issue of best practices in software development for the members of the bioinformatics community<sup>1-3</sup>. These practices include facilitating the discovery, deployment, and usage of tools, and several helpful solutions are available.

Tool discovery is facilitated by various online catalogs and registries<sup>4-6</sup>. The ELIXIR Tools and Data Services Registry, [bio.tools](#)<sup>7</sup>, describes bioinformatics software using extensive metadata descriptions, supported by the EDAM ontology<sup>8</sup>.

For software deployment, distribution systems are available<sup>9-13</sup> that let users locally install the tools that they need in convenient, portable and reproducible ways. Workbench and workflow systems such as Galaxy<sup>14,15</sup>, Taverna<sup>16</sup> or Chipster<sup>17</sup> allow the execution and composition of bioinformatics tools in integrated environments which aim at improved usability, interoperability and reproducibility. Finally, the Common Workflow Language<sup>18</sup> (CWL) is a recent project that defines a standardized and portable tool and workflow description format, usable across different platforms.

All of the above systems rely on components that provide the necessary information to describe, install, or run a specific piece of software. Gathering this information and formatting it into tractable tool descriptions is often a complex and time consuming task for developers. Indeed, it requires a deep knowledge of both the tool itself and the description format. A significant part of the metadata stored in the descriptions is, however, common to registries and workbench environments systems<sup>19</sup>, and strategies relying on a mapping between these different description formats can help avoid redundancy and mislabeling of tools [Figure 1](#)). The ReGaTE utility<sup>20</sup> illustrates this by using tool descriptions from Galaxy to publish available services on [bio.tools](#). Another

application is to facilitate workbench environment integration, by reusing tool descriptions from registries. Here we present “ToolDog” (Tool DescriptiOn Generator), an application that enables workbench integration for tools registered in the [bio.tools](#) registry.

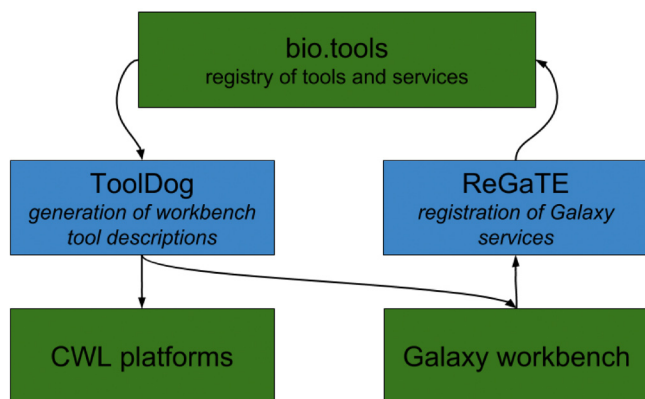
## Tool descriptions

Bioinformatics tools are described in various formats and levels of detail, befitting different systems and use-cases. A [bio.tools](#) entry provides tool descriptions for tool end-users, primarily for search and discovery purposes. The metadata provides a basic description including the tool type, what task it performs, the main input and output data, who created it, where it is available, and its license. This description, based on the [BiotoolsSchema model](#), can be accessed through the [bio.tools](#) API and retrieved in JSON format. Conversely, Galaxy and CWL tool descriptions must support tool discovery, execution, and integration into homogeneous environments. This requires an extensive description of their command line syntax (or other type of API). Galaxy tool descriptions are written in XML or YAML, and [the corresponding XSD](#) is available. CWL tool descriptions are described using the YAML-based [SALAD format](#).

All three of these tool description formats provide the possibility of specifying EDAM terms. In [bio.tools](#) this can be done directly. CWL supports these annotations through the addition of [bioschemas](#) mark-up, and Galaxy supports EDAM through specific tags mapping to its internal typing system<sup>21</sup>. The EDAM ontology helps with the description of the tools by providing a common vocabulary that includes terms to describe topics that specify which particular domains of bioinformatics the tool serves, operations that describe what the tool does, and data and formats that specify the type and format of the inputs and outputs.

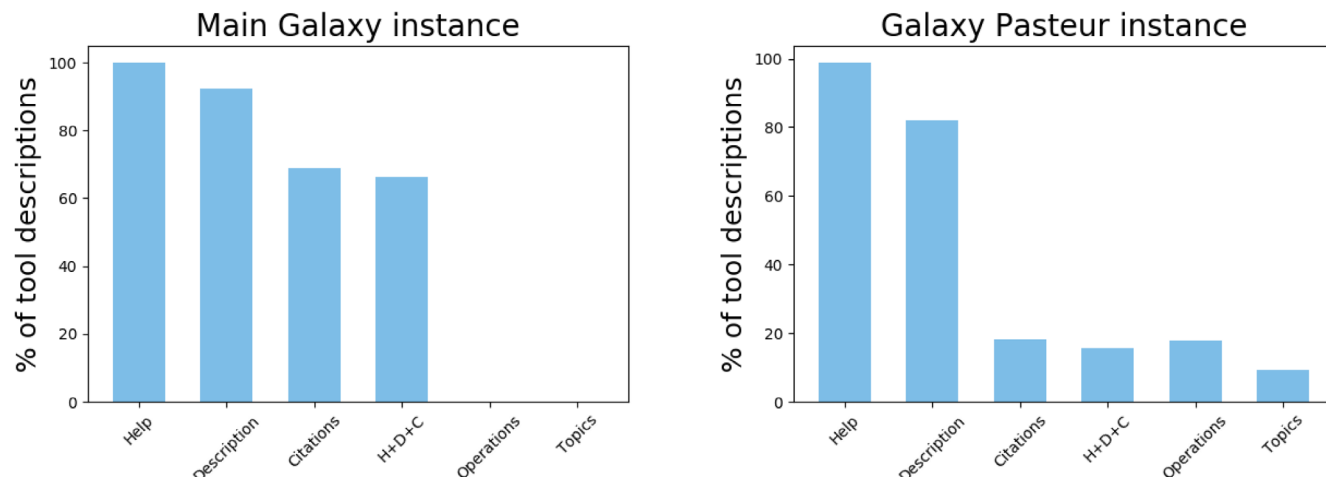
## Completeness of Workbench tool description

Tool descriptions for workbench systems are expensive to create and maintain, because they require exhaustive knowledge of both the described tool, and the syntax used for the description<sup>19</sup>. Consequently, tool descriptions are sometimes incomplete or out of date. For instance, in the case of Galaxy, the analysis of the [main server](#) and the server of the Institut Pasteur<sup>22</sup> shows that some tools are not adequately described (see [Figure 2](#)). Specifically, although most of the tools have a help section and a description, important elements such as citation information are often missing. The evolution of the Galaxy framework itself also generates a need for maintenance, through changes in the tool description format. With the recent addition of EDAM annotations tags in the format, tools had to be updated to support this new feature. The users of such graphical workbench platforms do not typically handle tool discovery and deployment tasks. Thus, detailed tool descriptions are fundamental, because they are the main source of information for the scientists who use them.



**Figure 1. Workbench Integration Enabler overview.** The objective is to integrate the [bio.tools](#) registry with workbench environments in two ways: (1) “ReGaTE”, a utility for *en masse* registration of services from Galaxy instances; (2) the “ToolDog” utility, to translate the description of any tool or service that is registered in [bio.tools](#), into the format required by the existing major workbench environments.

Different approaches exist to help improve the quality of the corpus of tool descriptions. (1) Tooling facilitates the creation and validation of the tool descriptions, using Planemo<sup>23</sup> in the case of Galaxy. (2) Community approaches such as the [Intergalactic Utilities Commission](#) design and promote best practices for the



**Figure 2.** Metadata coverage for Galaxy tool descriptions from (A) the main Galaxy instance (<https://usegalaxy.org>) and (B) the Institut Pasteur Galaxy instance (<https://galaxy.pasteur.fr>). The graphs show the percentage of tools possessing various metadata types: *Help*: usage instructions; *Description*: description of the tool to be displayed in the tool menu; *Citations*: tool citation information using either a DOI or a BibTeX entry; *H+D+C*: contains a help, description and citations section; *Operations*: description of the EDAM operation(s) performed; *Topics*: description of the EDAM topics covered. The total number of tools includes those which were successfully retrieved and analyzed (672 out of 1209 on Galaxy main, 351 out of 526 on Pasteur); not all available tools were retrieved - some because they are not available in a ToolShed, and some because we chose to retrieve only the latest version of each tool and discarded the earlier ones.

development of Galaxy tools. (3) Standardization efforts like CWL also reduce the maintenance work for tool descriptions by making them portable between different platforms.

ToolDog complements all of these approaches. It leverages the information available in bio.tools to simplify the integration of bioinformatics software into workbench environments.

## Methods

ToolDog is a command-line utility written in Python. It consists of two modules, which handle (1) the generation of a skeleton for the tool description, based on the analysis of the source code of the tool, and (2) the enrichment of the tool description, using the bio.tools metadata. The tool description generation pipeline (Figure 3) leverages bio.tools and includes both a module to generate a tool description using only the registry, as well as a module to enrich an existing tool description with information from the registry.

### Source code analysis

For a number of bioinformatics tools, a significant part of their description can be extracted from an analysis of the source code. The source code analysis module of ToolDog does this, currently only with python-based tools that use the *argparse* library for parsing command-line arguments. This module uses the *argparse2tool* package to retrieve the list of parameters and generate Galaxy or CWL tool description skeletons. To generate such skeletons, ToolDog runs a Docker software container that will download, install, analyze the source code, generate the tool description and then retrieve it. This strategy avoids the pollution of the local user's environment and provides a completely pre-configured, ready-to-use installation of ToolDog.

### Tool description enrichment

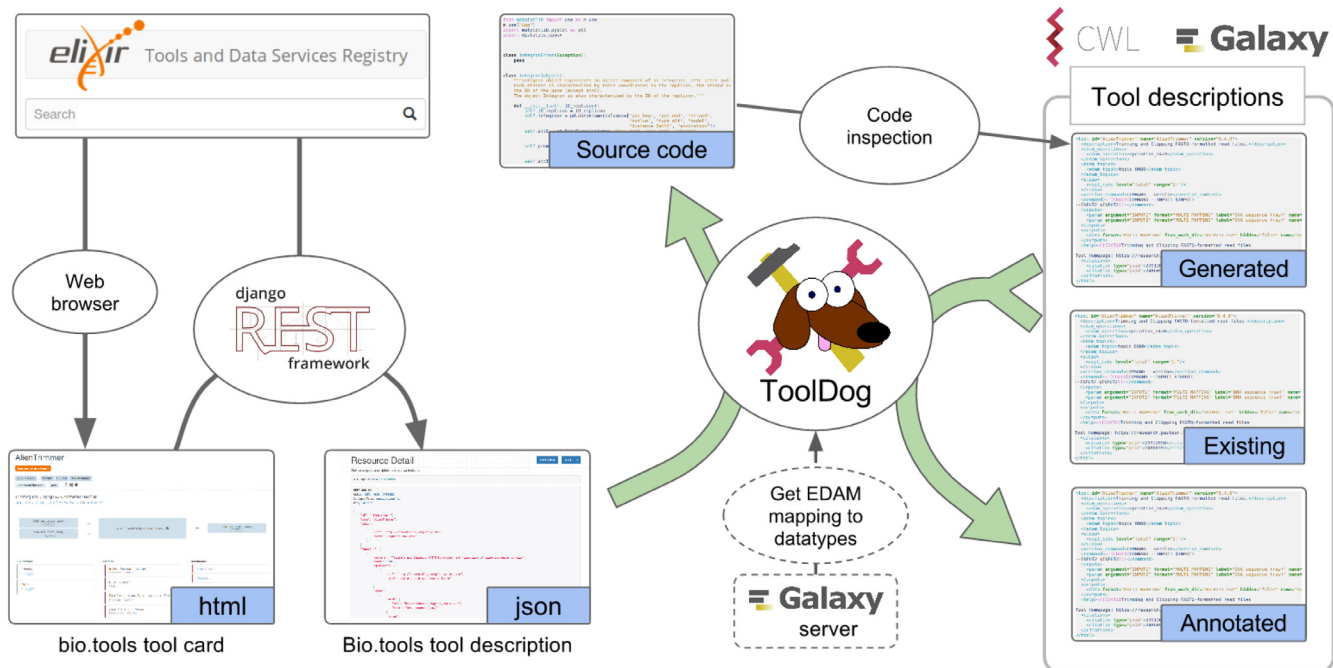
Galaxy and CWL tool descriptions, whether they were manually authored or automatically generated by source code analyses, can be improved by the description enrichment module. This retrieves additional metadata from the corresponding bio.tools entries, and fills in the missing information in the workbench tool description when available.

Internally, the input tool description is parsed into an object model of the tool. The metadata from bio.tools are then mapped onto this object model, which is later exported to Galaxy or CWL formats. Parsing and export capabilities of ToolDog leverage the *galaxyxml* or *cwlgen* libraries to import and export the updated descriptions.

## Results

### Generation of a tool description from a bio.tools entry

Here we illustrate the generation of a tool description with the example of *IntegronFinder*<sup>24</sup>, an analysis tool dedicated to the identification of integrons in bacterial genomes. Launching ToolDog in "generation mode" on the *IntegronFinder* entry in the bio.tools registry allows the generation of a significant portion of the tool description (Figure 4), either in CWL or Galaxy format. Some manual modifications (corrections + additions) are still necessary to complete the tool description and to make it functional. For instance, software requirements, which specify what software needs to be installed for the tool to run correctly, cannot be automatically generated, because this information is currently not available in bio.tools. Additionally, the mapping between inputs and the generated command line, as well as between outputs and the file names they refer to is not present.



**Figure 3.** ToolDog generates tool descriptors from bio.tools resources descriptions.

### Enrichment of an existing collection of tool descriptions

In addition to novel tool description generation, ToolDog can also perform the automated enrichment of existing tool descriptions with bio.tools metadata. To test this approach, we ran ToolDog on the tool descriptions available on the Galaxy main instance that lack EDAM annotations. All of the Galaxy descriptions from the main instance were retrieved, and mapped to bio.tools entries using the citation identifiers (DOI). The goal was to add EDAM terms describing the topic of application and the operation(s) performed by the tools. To avoid linking unrelated entries, we took a conservative approach, only mapping by default two entries when they referred to, and only to, the same publication. The results (Figure 5) show that whenever this linking can be reliably done, the enrichment can easily be performed, with a total of 217 Galaxy tool descriptions being enriched out of 224 being initially mapped to bio.tools. A detailed description of this analysis, including the original and annotated tool descriptions, is available at ([https://github.com/khillion/galaxyxml-analysis/annotate\\_usegalaxy](https://github.com/khillion/galaxyxml-analysis/annotate_usegalaxy)).

### Discussion

The ToolDog utility allows a developer to generate new tool descriptions for tools which are compatible with the code analysis module, and reuse the metadata provided by bio.tools to enrich existing tool descriptions. There are some limitations to this approach:

1. The “plugin” libraries used for code analysis are specific to the programming languages, libraries or framework

used to build the command line interface. To this date, they don’t cover most of these.

2. The generation of the tool descriptions through code analysis must assume certain coding practices, such as the use of specific functions to define input or output parameters, which are not uniformly adopted.
3. Some of the input/output operations performed by some programs are a lot more difficult to detect through code analysis because they are typically not included in command line parsing frameworks, such web service and database queries and submissions, or in place file modifications.

The automated enrichment of existing tool descriptions provides a convenient way to improve them, especially if they lack most of the metadata provided by bio.tools. Performing this enrichment efficiently *en masse*, however, would require the wide adoption of an identification system for bioinformatics software. This mechanism would allow to avoid the complex and sometimes ambiguous mapping procedures based on publication identifiers we performed when testing it on the Galaxy tools. A recent update to bio.tools has added stable and unique tool identifiers, based on registered tool names, yielding persistent references to tools, for example <https://bio.tools/signalp>. Future work will make use of these identifiers to improve the generation of tool descriptions. For instance, linking of the bioconda and biocontainers repositories to bio.tools will enable ToolDog to generate software requirements compatible with workbench platforms<sup>25</sup>.

```

$tooldog -c integron_finder/1.5.1

1  #!/usr/bin/env cwl-runner
2  cwlVersion: v1.0
3  inputs:
4    replicon:
5      doc: Path to the replicon file (in fasta format), eg - path/to/file.fst or file.fst
6      type: string
7      inputBinding:
8        position: 1
9      [...]
10 baseCommand:
11 - integron_finder
12 class: CommandLineTool

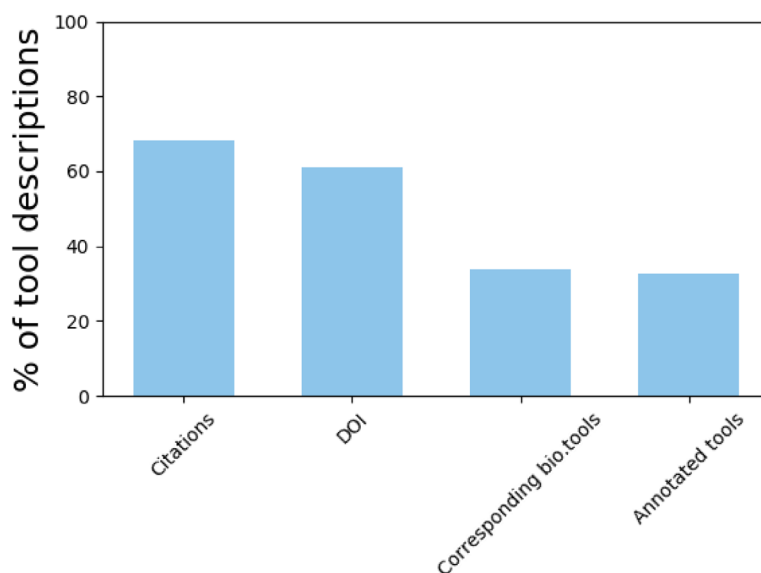
$tooldog -g integron_finder/1.5.1

1  <tool id="integron_finder" name="integron_finder" [...]>
2    <description>A tool to detect Integron in DNA sequences.</description>
3    <edam_operations>
4      <edam_operation>operation_3430</edam_operation>
5      [...]
6    </edam_operations>
7    <command><![CDATA[integron_finder
8      #if $positional_1 and $positional is not None:
9      $positional_1
10     #end if
11     [...]
12   > $default]]></command>
13   [...]
14   <inputs>
15     <param area="false" argument="positional_1" label="Path to the replicon file (in fasta format),
16       ↳ eg: path/to/file.fst or file.fst" name="positional_1" type="text"/>
17     [...]
18   </inputs>
19   <outputs>
20     <data format="txt" hidden="false" name="default"/>
21   </outputs>
22   <help><![CDATA[
23   What it is ?
24   =====
25
26   A tool to detect Integron in DNA sequences
27
28   External links:
29   =====
30
31   - Tool homepage_
32   - bio.tools_ entry
33
34   .. _homepage: https://github.com/gem-pasteur/Integron_Finder
35   .. _bio.tools: https://bio.tools/tool/integron_finder]]></help>
36   < citations>
37     < citation type="doi">10.1093/nar/gkw319</ citation>
38   </ citations>
39 </ tool>

```

**Figure 4.** Output of the run of ToolDog using the bio.tools entry of IntegronFinder to generate the corresponding CWL and Galaxy tool descriptions.





**Figure 5. Tool descriptions automated mapping and enrichment.** Out of 665 retrieved tool descriptions, 399 have a DOI and 224 of these descriptions could be mapped to a bio.tools entry. 217 tool descriptions have been successfully annotated using ToolDog (*Citations*: presence of tool citation information; *DOI*: tool citation information described using a DOI; *Corresponding bio.tools*: tool descriptions with a corresponding bio.tools entry retrieved using the DOI; *Annotated tools*: tool descriptions successfully annotated with ToolDog).

## Conclusions

During the last years, integration of various tools has been eased by the use of workbench systems such as Galaxy, and frameworks using the Common Workflow Language. Still, it remains time consuming and not straightforward to adapt resources to such environments. ToolDog lays the foundation for future work, that will provide a Workbench Integration Enabler for the bio.tools registry as an online service. Furthermore, integration with Planemo, the main utility to develop Galaxy and CWL tools, will be further developed in order to make the simple, bio.tools-based metadata enrichment of ToolDog available to the widest possible audience.

## Data availability

The scripts and results of the analysis performed to motivate and test our approach are available at: <https://github.com/khillion/galaxyxml-analysis>, and are archived at the time of publication at: <https://doi.org/10.5281/zenodo.1038005><sup>26</sup>.

## Software availability

The ToolDog software is available at: <https://pypi.python.org/pypi/tooldog>

The source code is available at: <https://github.com/bio-tools/tooldog>

Archived source code as at the time of publication: <https://doi.org/10.5281/zenodo.1037909><sup>27</sup>

Software license: MIT License.

## Competing interests

No competing interests were disclosed.

## Grant information

ELIXIR-EXCELERATE is funded by the European Commission within the Research Infrastructures Programme of Horizon 2020 [676559].

## Acknowledgments

Jon Ison acknowledges the support of the Danish ELIXIR Node. Kenzo-Hugo Hillion and Hervé Ménager wish to thank Fabien Mareuil, Olivia Doppelt-Azeroual, Bertrand Néron from the Institut Pasteur, as well as Daniel Blankenberg (Cleveland Clinic) and John Chilton (Galaxy Project) for their technical advice during the development. Anton Khodak wishes to thank his Google Summer of Code mentor Roman Valls Guimera (University of Melbourne), who promoted the idea of argparse2tool and supervised his internship.



## References

1. Artaza H, Chue Hong N, Corpas M, *et al.*: **Top 10 metrics for life science software good practices [version 1; referees: 2 approved]**. *F1000Res*. 2016; 5: pii: ELIXIR-2000.  
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
2. Jiménez RC, Kuzak M, Alhamdoosh M, *et al.*: **Four simple recommendations to encourage best practices in research software [version 1; referees: 3 approved]**. *F1000Res*. 2017; 6: pii: ELIXIR-876.  
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
3. Silva LB, Jimenez RC, Blomberg N, *et al.*: **General guidelines for biomedical software development [version 1; referees: 2 approved]**. *F1000Res*. 2017; 6: 273.  
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
4. Bhagat J, Tanoh F, Nzuobontane E, *et al.*: **BioCatalogue: a universal catalogue of web services for the life sciences**. *Nucleic Acids Res*. 2010; 38(Web Server issue): W689–W694.  
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
5. Henry VJ, Bandrowski AE, Pepin AS, *et al.*: **OMICtools: an informative directory for multi-omic data analysis**. *Database (Oxford)*. 2014; 2014: pii: bau069.  
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
6. Tan P, Zhou Y, Huang X, *et al.*: **AZTEC: A cloud-based computational platform to integrate biomedical resources**. In *2017 IEEE 33rd International Conference on Data Engineering (ICDE)*. IEEE, 2017.  
[Publisher Full Text](#)
7. Ison J, Rapacki K, Ménager H, *et al.*: **Tools and data services registry: a community effort to document bioinformatics resources**. *Nucleic Acids Res*. 2016; 44(D1): D38–D47.  
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
8. Ison J, Kalas M, Jonassen I, *et al.*: **EDAM: an ontology of bioinformatics operations, types of data and identifiers, topics and formats**. *Bioinformatics*. 2013; 29(10): 1325–1332.  
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
9. Möller S, Krabbenhöft HN, Tille A, *et al.*: **Community-driven computational biology with debian linux**. *BMC Bioinformatics*. 2010; 11(Suppl 12): S5.  
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
10. da Veiga Leprevost F, Grüning BA, Alves Afrits S, *et al.*: **BioContainers: an open-source and community-driven framework for software standardization**. *Bioinformatics*. 2017; 33(16): 2580–2582.  
[PubMed Abstract](#) | [Publisher Full Text](#)
11. Moreews F, Sallou O, Ménager H, *et al.*: **BioShaDock: a community driven bioinformatics shared Docker-based tools registry [version 1; referees: 2 approved]**. *F1000Res*. 2015; 4: 1443.  
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
12. O'Connor BD, Yuen D, Chung V, *et al.*: **The Dockstore: enabling modular, community-focused sharing of Docker-based genomics tools and workflows [version 1; referees: 2 approved]**. *F1000Research*. 2017; 6: 52.  
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
13. Dale R, Grüning B, Sjödin A, *et al.*: **Bioconda: A sustainable and comprehensive software distribution for the life sciences**. *bioRxiv*. 2017.  
[Publisher Full Text](#)
14. Blankenberg D, Von Kuster G, Coraor N, *et al.*: **Galaxy: a web-based genome analysis tool for experimentalists**. In Frederick M. Ausubel, Roger Brent, Robert E. Kingston, David D. Moore, JG. Seidman, John A. Smith, and Kevin Struhl, editors, John Wiley & Sons, Inc., Hoboken, NJ USA, *Curr Protoc Mol Biol*. 2010; Chapter 19: Unit 19.10.1-21.  
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
15. Giardine B, Riemer C, Hardison RC, *et al.*: **Galaxy: a platform for interactive large-scale genome analysis**. *Genome Res*. 2005; 15(10): 1451–1455.  
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
16. Wolstencroft K, Haines R, Fellows D, *et al.*: **The Taverna workflow suite: designing and executing workflows of Web Services on the desktop, web or in the cloud**. *Nucleic Acids Res*. 2013; 41(Web Server issue): W557–W561.  
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
17. Kallio MA, Tuimala JT, Hupponen T, *et al.*: **Chipster: user-friendly analysis software for microarray and other high-throughput data**. *BMC genomics*. 2011; 12(1): 507.  
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
18. Amstutz P, Crusoe MR, Tijanić N, *et al.*: **Common workflow language, v1. 0**. *figshare*. 2016.  
[Publisher Full Text](#)
19. Ménager H, Kalaš M, Rapacki K, *et al.*: **Using registries to integrate bioinformatics tools and services into workbench environments**. *International Journal on Software Tools for Technology Transfer*. 2016; 18(6): 581–586.  
[Publisher Full Text](#)
20. Doppelt-Azeroual O, Mareuil F, Deveaud E, *et al.*: **ReGaTE: Registration of Galaxy Tools in Elixir**. *Gigascience*. 2017; 6(6): 1–4.  
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
21. Afgan E, Baker D, van den Beek M, *et al.*: **The galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2016 update**. *Nucleic Acids Res*. 2016; 44(W1): W3–W10.  
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
22. Mareuil F, Doppelt-Azeroual O, Ménager H: **A public galaxy platform at pasteur used as an execution engine for web services**. 2017.  
[Publisher Full Text](#)
23. Chilton J, Guerler A: **Planemo: a scientific workflow sdk**. 2016.  
[Publisher Full Text](#)
24. Cury J, Jové T, Touchon M, *et al.*: **Identification and analysis of integrons and cassette arrays in bacterial genomes**. *Nucleic Acids Res*. 2016; 44(10): 4539–4550.  
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
25. Grüning B, Chilton J, Köster J, *et al.*: **Practical computational reproducibility in the life sciences**. *bioRxiv*. 2017.  
[Publisher Full Text](#)
26. Hillion KH, just another pesky drone: **khillion/galaxyxml-analysis: v1.0.2 for F1000 submission (Version v1.0.2)**. *Zenodo*. 2017.  
[Data Source](#)
27. Hillion KH, just another pesky drone, Kuzmin I, *et al.*: **bio-tools/ToolDog: v0.3.4 for F1000 submission (Version v0.3.4)**. *Zenodo*. 2017.  
[Data Source](#)

# Open Peer Review

Current Referee Status:



Version 1

Referee Report 16 January 2018

doi:10.5256/f1000research.14069.r28595



**Brian O'Connor** 

Computational Genomics Platform, UCSC Genomics Institute, University of California, Santa Cruz, Santa Cruz, CA, USA

"Using bio.tools to generate and annotate workbench tool descriptions" is an article that describes a tool descriptor program known as ToolDog. It was designed to generate Galaxy XML or CWL from particular bioinformatics tool source code as well as metadata annotations on bio.tools. The idea is great, since the issue is a real one in the community. Namely, there are a lot of tools out there but typically they lack descriptors in Galaxy or CWL format. And this makes it harder to use in "workbench" and workflow systems. Creating a tool that tool authors can use to help create descriptors is awesome. Source is available in GitHub and the tool can be installed via pip.

## Feedback/Questions

1. Can the authors rename the article? I think it should include ToolDog in the article title.
2. What are the plans for other languages (if any)? Do the authors see ToolDog as something that others will extend for, say, WDL generation?
3. I think it would be interesting to hear more about future plans. Specifically, how will the authors expand this to a Workbench Integration Enabler? Do they see this as being an automated process? How will they leverage the work of bioconda and biocontainers (they did mention this briefly) and will the goal be to generate CWL/GalaxyXML for everything in bio.tools + bioconda/biocontainers?
4. Alternatively, if the goal not to automatically export CWL/GalaxyXML for everything in bio.tools, is it, instead, to provide a tool for tool authors to use when building their tool to jumpstart their descriptor creation? Some clarification on the intended audience I think would be helpful.
5. The authors described generating CWL/Galaxy XML for IntegronFinder. Did they try other tools and, if so, how successful was that? What about generation in bulk?
6. Can they comment on what a tool author should do with the generated CWL or Galaxy XML? They mention in the results that some work is required to make the tool run correctly. Is the tool author then suggested to check in the CWL/Galaxy XML to their source repo and maintain it? What is the recommendation here?

**Is the rationale for developing the new software tool clearly explained?**

Yes

**Is the description of the software tool technically sound?**

Yes

**Are sufficient details of the code, methods and analysis (if applicable) provided to allow replication of the software development and its use by others?**

Yes

**Is sufficient information provided to allow interpretation of the expected output datasets and any results generated using the tool?**

Yes

**Are the conclusions about the tool and its performance adequately supported by the findings presented in the article?**

Yes

**Competing Interests:** No competing interests were disclosed.

**I have read this submission. I believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard.**

Referee Report 10 January 2018

doi:10.5256/f1000research.14069.r28567



**Manuel Corpas** 

Cambridge Precision Medicine, Cambridge, UK

The article 'Using bio.tools to generate and annotate workbench tool descriptions' describes the software tool ToolDog. ToolDog improves the interoperability of bio.tool-deposited entries within workbenches by converting their descriptions into formats that are compatible with workflow standards.

ToolDog is a convenient addition to the existing capabilities for the integration of bio.tools entries with workbench environments.

I found Figure 2 particularly interesting, describing the metadata coverage descriptions from two of the main Galaxy servers. Do you have the raw data with which this figure was created? It would be good to have it openly shared. Figure 2 illustrates the problem of the significant lack of completeness in crucial metadata descriptions of Galaxy tools.

My main recommendation for this article would be to provide a step-by-step guide on how to run ToolDog using a self-contained example. I feel unable to test the tool because I do not know how to download the metadata from a bio.tools entry and need to set up my python environment, download the code and make it work. This article, although it is geared toward a programmer audience, it would be hard to test/reproduce for someone who is not a seasoned python programmer. I would thus recommend a beginner's guide for those of us who are not so technical.

Other than that, I am glad to see all the source code adequately deposited both in github and Zenodo for the snapshot image for this publication. The MIT license is also commendable as it allows free reuse and modification.

Finally some minor corrections:

- Link in the first paragraph of the results section 'of a significant portion of the tool description' is broken
- Link on the second paragraph of the results section '[https://github.com/khillion/galaxyxml-analysis/annotate\\_usegalaxy](https://github.com/khillion/galaxyxml-analysis/annotate_usegalaxy)' is broken
- Discussion section bullet point #3 'such web service' ==> such as web services

**Is the rationale for developing the new software tool clearly explained?**

Yes

**Is the description of the software tool technically sound?**

Yes

**Are sufficient details of the code, methods and analysis (if applicable) provided to allow replication of the software development and its use by others?**

Yes

**Is sufficient information provided to allow interpretation of the expected output datasets and any results generated using the tool?**

Yes

**Are the conclusions about the tool and its performance adequately supported by the findings presented in the article?**

Yes

**Competing Interests:** No competing interests were disclosed.

**Referee Expertise:** Computational Genomics

**I have read this submission. I believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard.**

Referee Report 18 December 2017

doi:10.5256/f1000research.14069.r28565



**Christopher J. Fields** 

High-Performance Biological Computing Group, Roy J. Carver Biotechnology Centre, University of Illinois at Urbana-Champaign, Urbana, IL, USA

The paper presents a very nice overview on how ToolDog is used to (1) generate new tool descriptors for Galaxy and CWL from code analysis, and (2) improve documentation for current tools from the bio.tools registry. This provides a valuable service to the bioinformatics community and in particular to ensuring

that tooling information is consistently described but also updatable. In my opinion this should be accepted, with some minor suggested revisions.

Speaking of 'suggestions':

1. The current title 'Using bio.tools to generate and annotate workbench tool descriptions' suggests the paper will talk more generally about bio.tools, whereas the text focuses primarily on the specific component ToolDog. The title should be modified to reflect this.
2. The graphs in Fig.2 would be more effective if they were displayed in an integrated manner (single bar chart?), so that the improvements that ToolDog makes are more easily compared to one another.
3. The discussion about the challenges in autogenerating tool documentation (language, code practices, etc), in the discussion, are spot-on. However not much is discussed on if / how ToolDog might address some of these challenges, though there are suggestions on how to more readily map existing tool descriptions to add to or update. Maybe this could be elaborated on, even if it's indicating the problems may not be easily overcome?
4. I'm wondering whether the information in Fig. 5 might be better displayed (or augmented) as a before / after comparison to more readily demonstrate how ToolDog could automatically improve tool descriptions. Another option is whether this information could be somehow connected to the data in Fig. 2 to show how ToolDog improves the overall documentation.

**Is the rationale for developing the new software tool clearly explained?**

Yes

**Is the description of the software tool technically sound?**

Yes

**Are sufficient details of the code, methods and analysis (if applicable) provided to allow replication of the software development and its use by others?**

Yes

**Is sufficient information provided to allow interpretation of the expected output datasets and any results generated using the tool?**

Yes

**Are the conclusions about the tool and its performance adequately supported by the findings presented in the article?**

Partly

**Competing Interests:** No competing interests were disclosed.

**Referee Expertise:** Computational biology

**I have read this submission. I believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard.**

Referee Report 15 December 2017

doi:10.5256/f1000research.14069.r28566



**Michael L. Heuer**  1,2

<sup>1</sup> Department of Bioinformatics Research, National Marrow Donor Program (NMDP), Minneapolis, MN, USA

<sup>2</sup> AMPLab, University of California, Berkeley, CA, USA

From the point of view of a software tool author, it is not a simple task to provide high-quality metadata and software tool descriptions. Any tooling that supports DRY (don't repeat yourself) in this regard is most welcome.

The authors describe a path from the bio.tools bioinformatics software registry, which uses a rich metadata schema for syntax, the EDAM ontology for semantics, and strongly written guidelines to ensure high-quality entries, to tool descriptions for the Galaxy workbench and in Common Workflow Language (CWL) for use on various workflow execution environments.

Much of the metadata in the tool descriptions is generated by the ToolDog utility from an entry in bio.tools, ensuring proper mapping between metadata concepts. This would be a great help when bootstrapping Galaxy and CWL support for a new software tool. The authors also describe and implement a use case for enriching existing tool descriptions.

I am curious if there are practical benefits to enriching tool descriptions with EDAM ontology terms, in addition to quality of metadata from using well defined terms from a controlled vocabulary?

The source code is available at the Github link provided and is licensed MIT License as stated in the paper. I appreciate that the scripts and results of the analysis are archived as well.

**Is the rationale for developing the new software tool clearly explained?**

Yes

**Is the description of the software tool technically sound?**

Yes

**Are sufficient details of the code, methods and analysis (if applicable) provided to allow replication of the software development and its use by others?**

Yes

**Is sufficient information provided to allow interpretation of the expected output datasets and any results generated using the tool?**

Yes

**Are the conclusions about the tool and its performance adequately supported by the findings presented in the article?**

Yes

**Competing Interests:** No competing interests were disclosed.

**Referee Expertise:** Bioinformatics, big data genomics, immunogenomics

**I have read this submission. I believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard.**

---

The benefits of publishing with F1000Research:

- Your article is published within days, with no editorial bias
- You can publish traditional articles, null/negative results, case reports, data notes and more
- The peer review process is transparent and collaborative
- Your article is indexed in PubMed after passing peer review
- Dedicated customer support at every stage

For pre-submission enquiries, contact [research@f1000.com](mailto:research@f1000.com)

**F1000Research**